TCP/IP—A Dependable Networking Infrastructure

by Carl Malamud

Think of TCP/IP as the sedate and practical Oldsmobile of modern protocol stacks. It is old, dependable and not particularly flashy. At the same time, its popularity makes it more like a Honda (it sells itself). How did TCP/IP (transmission control protocol/internetwork protocol) become so popular? Two words can explain its success: It works. And an old adage, "If it isn't broken, don't fix it," explains its longevity.

We've heard for years that TCP/IP was just a temporary solution—a transition until OSI (Open Systems Interconnection) would take over for open systems. Vendors of proprietary systems—DECnet, SNA, NetWare and others—pay lip service to TCP/IP but then go on to argue that their own products are better.

But TCP/IP protocols prevail. Their basic services get wide use, and value-added services, such as Sun's Network File System, continue to appear. Indeed, the future of TCP/IP looks very good even in the face of more modern protocol suites like those offered by OSI.

Widespread Use

TCP/IP reveals its staying power when many different networks share a common base of services. A few years ago, vendors thought TCP/IP would die. They talked then about moving to OSI, which they considered the ultimate in open systems. Now most of them, recognizing TCP/IP's large installed base with its hundreds of thousands of nodes, speak of the coexistence of TCP/IP and OSI.

Because of that installed base, new applications continue to appear. One of the biggest successes, for example, has been the Simple Mail Transfer Protocol (SMTP). Originally considered, like TCP/IP, to be only an interim solution, the service has clearly won out over other approaches.

Among the likely reasons for TCP/IP's popularity is that there have been many prototype implementations of the protocols to test as part of the standards process. This, in turn, has further been aided by a very open process for participating in standards-making decisions.

What's more, distribution of the standards is free. This is in sharp contrast to OSI, where many protocols become standards without ever being implemented, and where information can run as much as $10 per page (since the protocols are copyrighted).

In the Beginning

TCP/IP evolved from a mid-'70s research project to find a way to link a few computers throughout the United States. Funded by the Defense Advanced Research Projects Agency (DARPA), the goal was to let researchers at one site access a computer at another site.

The research caught on quickly and led to ARPANET (a wide area network established by DARPA). Based on leased lines and packet-switched networks, ARPANET underwent explosive growth. In 1980, DARPA started switching ARPANET over to a newly developed protocol suite: TCP/IP.

Hundreds of thousands of TCP/IP-based nodes sprang up, linked over local and wide area networks. The largest and most popular wide area network for TCP/IP is what is now known as the Internet. Also, the large number of nodes and wide area networks put TCP/IP to an early and demanding trial. Consequently, the TCP/IP protocols have been tested under heavy fire in all kinds of networking environments. TCP/IP's performance has also contributed to its success.

Yet another reason for TCP/IP's popularity is that its protocols are in the public domain. Anyone can write a TCP/IP implementation without getting a license or performing any other administrative procedures.

Unlike copyrighted standards like OSI, TCP/IP is available to anyone at very low cost. According to Jon Postel, editor of the official Internet standards, public access has led to important contributions by software startups, programmers and even students, who have readily added value throughout the TCP/IP-based networking environment.

Having TCP/IP in the public domain also made it attractive to network implementers. As a result, implementations cropped up on a variety of computers. Particularly important was the incorporation of TCP/IP into the Berke-

PHOTO COURTESY WIDE WORLD PHOTOS

ley Software Distribution of Unix. Because most university computer science departments were running the Berkeley version of Unix, TCP/IP was made available to them by default.

How It Works

The TCP/IP standards build on existing technology. Ethernet, X.25, HDLC, FDDI and Token-Ring are all data link standards—they are methods of getting a packet of data from one computer to another.

IP, the Internet Protocol, is a way of linking subnetworks. In the terminology of the OSI Reference Model, it is a network layer protocol. The IP module takes a packet from one data link, consults a routing table and forwards it to another data link. Moreover, collections of data links are organized into networks.

For this scheme to work, each node (workstation, server, router, gateway) must have a unique address. Therefore, IP addresses are divided into four decimal parts: for example, 128.97.49.5. Each part of the address refers to a level in an address hierarchy.

The four parts of an address go from the most general to the most specific level. In the example, 128 is the broadest part of the address. It can be compared to "USA" on a piece of paper mail. The 5 is the most specific part of the address—like the street address of a house—and is the actual ID of a specific node.

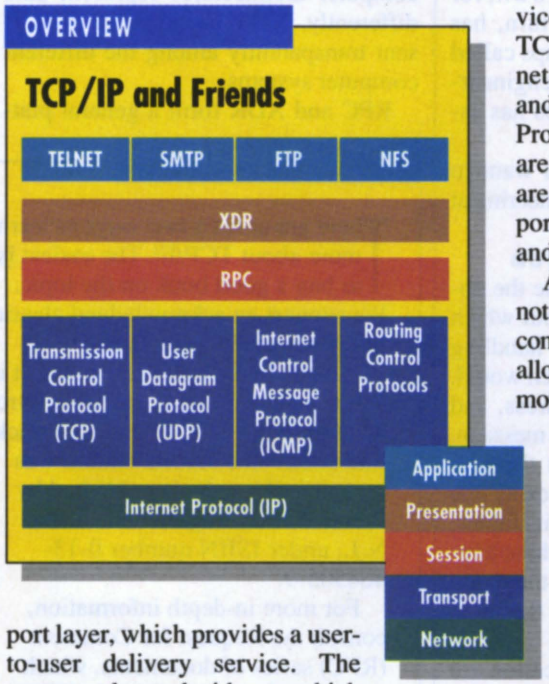
Knowing how to get a packet to any node on a given network is only half of what an IP module does. It also knows how to get a packet to other networks, which it does through a router (either a specific piece of hardware dedicated to routing, or a workstation or server acting as one). The router connects two TCP/IP-based networks. It also tells the IP modules on other nodes which networks they are connected to—a process that is handled by IP's routing control protocols.

Collections of nodes on a particular network exchange the routing-control information necessary to ensure that packets can be properly forwarded. For example, node A tells other nodes on network X, "I can reach networks Y and Z." When a node on network X sees a packet destined for network Y, it sends that packet to node A, which then acts as the router, forwarding the packet to the appropriate network.

Transport Protocols

The IP protocol accepts a packet and, by forwarding it to other IP modules on a network or across networks, eventually delivers the packet to its destination. Maybe. IP is a so-called datagram or connectionless service. For each packet it sends, it provides only a best-effort delivery service. Maybe the packet will get there, maybe it won't.

IP provides a computer-to-computer delivery service. Built on top of the network layer that IP services is the transport layer, which provides a user-to-user delivery service. The transport layer decides to which user an incoming packet from the network layer belongs. Additionally, it may feature value-added services such as guaranteed delivery.



There are two basic transport service providers in the TCP/IP protocol suite: UDP (user datagram protocol) and TCP. UDP serves an end user but does not guarantee that data will be delivered. UDP preserves the connectionless datagram orientation of the underlying IP layer, and thus provides a very low-overhead transport service.

In contrast, TCP is known as a stream-oriented service provider. It guarantees that all data submitted to TCP at one end of the network will emerge error-free at the other end, and in the order it was sent.

TCP is a connection-oriented service that delivers data reliably. However, because TCP sets up a connection, it carries more overhead and is somewhat less efficient than UDP, particularly when dealing with quick and short communications.

Which is preferred? It's up to the services or users on the network. Some services can use both transport connections. Others require TCP to ensure that data is not lost.

Basic Services

In short, TCP, UDP, IP and other service protocols work together to deliver data from one user to another. The user of the service is an application program.

While there are different services available to run on top of TCP/IP, three basic ones are telnet, file transfer protocol (ftp), and the Simple Mail Transfer Protocol (SMTP). The services are simple and, for that reason, are easy to implement. They support a wide range of computers and operating systems.

An ftp is a way to copy a file, but not individual records, from one computer node to another. Telnet allows an interactive logon to a remote computer. It provides fairly simple terminal emulation and limited graphics. SMTP, as its name implies, sends mail messages.

Supplemental Services

The Internet has a variety of other protocols available. Some are new and just finding their way into TCP/IP networks. Others are experimental and may never be widely used. Some of these services provide the basic building blocks for the next generation of TCP/IP services. The domain naming service (DNS), the network time protocol and network authentication are examples of those building blocks.

DNS provides a means of keeping track of computers and the people who use them. If a computer's Internet address is known, we can use ftp to transfer files to and from it, and telnet to log into it. And if a person's logon ID at a certain computer is known, mail can be sent to that user.

DNS provides a way of distributing the administration of names and the IP addresses that go with them. When a mail program sees a name, it consults the DNS to learn its Internet address. The result is that names are separated from addresses, allowing great flexibility in configuring a large network. Flex-

ibility arises because control can be delegated to lower layers, and neither applications nor users need to understand complex IP addresses.

Further, though the IP addressing scheme is a flat address space, DNS provides a hierarchical view of the network, allowing managers of top-level domains to have administrative authority to assign names down the hierarchy. For example, in a commercial domain, known as `com`, the administrator for the commercial domain has allocated a subdomain called `sun`, for Sun Microsystems. Sun, In turn, has split this subdomain into groups called `corp` (corporate) and `eng` (engineering), and each of these groups has assigned user names.

For example, let's say you want to send mail to Joe Miller in engineering at Sun. His address might be:

`jmiller@eng.sun.com`

Your mail system would use the domain naming service to find out which computer is responsible for handling mail for `sun.com`. This search would, in turn, yield an Internet address, and SMTP would then transfer the message. Once the message arrives at the machine corresponding to `sun.com`, that machine would either search its tables, or ask DNS, to see which machine handles `eng.sun.com`. From there, the message would be sent over the local network to a particular user.

Time service and authentication are two services starting to come online. Time service allows different computers to synchronize their clocks, which is essential in an environment in which updates to data may come from many sources.

Authentication keeps computers up to date and informed of each other's identities. Because this information is crucial in a world of hundreds of thousands of interconnected computers, the Internet Engineering Task Force (IETF) is working quickly to advance this area.

Value-added Services

The ftp, telnet and SMTP services are all examples of fairly primitive services that run very well in a wide area environment. TCP/IP, though, is not limited to wide area networks. It can serve just as well as the basis for a workgroup LAN.

To make TCP/IP more functional in a LAN-based environment, Sun developed extensions to the basic model. These extensions are known as the Open Network Computing (ONC) family of protocols and consist of two basic services: remote procedure call, or RPC, and external data representation, referred to simply as XDR.

RPC provides the means for two remote computers to coordinate requests with each other in order to be able to execute particular procedures on one computer and return results back over the network to the other. Because various computer architectures represent data differently, XDR ensures that data is sent transparently among the different computer systems.

RPC and XDR form a general platform for developing applications in a distributed computing environment.

In fact, several companies have already based their software on this platform. Frame, for example, has built a "live links" feature into its publishing software. The feature lets pieces of a document reside on different computing systems.

The simplest reason for using RPC and XDR as a development platform for distributed is simple—the protocols are in the public domain. For those who don't want to start from scratch, source code is available, and can be licensed at low cost from a number of different vendors.

The result, as with the TCP/IP protocols, is a wide acceptance of both RPC and XDR. Over a million computers are now running the protocols, which support all the major operating systems.

Sun's Network File System (NFS), which is built on top of RPC and XDR, further helps the popularity of both of these protocols. NFS, also freely available, allows a file system on one computer to appear as a local file system to another. A workstation can "mount" files from a Cray, IBM mainframe, VAX or other platforms and have them all appear as local files.

When a protocol suite such as Network File System becomes widely implemented, heterogeneous file systems work across a range of computing architectures.

With its wide acceptance and popularity still going strong—just as the reliable family sedan continues to roll steadily along—TCP/IP still has plenty more miles left to go before it is replaced by OSI. ■

Carl Malamud writes professional reference books. He recently completed Analyzing DECnet/OSI Phase V, *to be published by Van Nostrand Reinhold.*

FOR MORE INFORMATION

There are at least two ways to learn more about TCP/IP. The easiest is to buy a good book on the topic. We suggest an acknowledged classic, *Internetworking with TCP/IP, Volume 1*, by Douglas E. Comer. It is a thorough and highly readable introduction to the TCP/IP protocol stack. The book—the second edition was recently released—is published by Prentice Hall, Englewood Cliffs, N.J., under ISBN number 0-13-468505-9.

For more in-depth information, consult the Request for Comment (RFC) series of documents, which details each of the protocols. RFCs can be obtained by electronic mail from the NSFNET Network Service Center. Here's the procedure:

• First, you need to get your message to the center. If you're on the Internet, you simply send mail to:

info-server@sh.cs.net

If you're on another system, use their Internet gateway.

For example, from MCI address mail as follows:

TO: CIC (EMS)
EMS: Internet
MBX:info-server@sh.cs.net

And if you're on CompuServe, send to:

>Internet:info-server@sh.cs.net

• When sending your message, you can leave the subject line blank.

• In the body of the message, include two lines:

REQUEST: RFC
TOPIC: HELP

• Finally, you'll get a message telling you how to get specific documents.

A good start is RFC1118, the *Hitchhiker's Guide to the Internet.*